Journal of Computer Allied Intelligence
ISSN: 2584-2676
Vol.02(04), August 2024, pp.70-81

Research Article

Robot Path Planning and Tracking with the Flower Pollination Search Optimization Algorithm

P. Brundavani[1,*] and Y. Avanija[2]
[1] Associate Professor, Department of ECE, Ramireddy Subbaramireddy Engineering College Kavali, SPSR Nellore, A.P, 524142, India.
[2] Assistant Professor, Department of ECE, PVKKIT, Ananthapuramu, A.P,515002, India.
[*]Corresponding Author: P. Brundavani. Email: brundavenky777@gmail.com
Received: 12/07/2024; Revised: 28/07/2024; Accepted: 20/08/2024; Published: 31/08/2024.
DOI: https://doi.org/10.69996/jcai.2024020

Abstract: Robot path planning and tracking involve two critical aspects of autonomous navigation systems: determining an optimal route for the robot to follow and ensuring that it accurately adheres to this path in real-time. Path planning focuses on generating a feasible and efficient trajectory from a starting point to a destination while considering obstacles, dynamic environments, and constraints. This paper investigates the effectiveness of the Flower Pollination Search Optimization (FPSO) algorithm for robot path planning and compares its performance with traditional algorithms such as A* Algorithm, Dijkstra, and Rapidly-exploring Random Tree (RRT). The FPSO algorithm was evaluated across three distinct scenarios, demonstrating superior performance in terms of path length, computation time, and path smoothness. In Scenario 1, FPSO achieved an optimized path length of 10.5 meters with a computation time of 3.2 seconds, a path smoothness score of 8.9, and a path efficiency of 95%. In Scenario 2, FPSO resulted in a path length of 12.3 meters and a computation time of 3.5 seconds, with a smoothness score of 8.7 and an efficiency of 93%. Scenario 3 showed FPSO's best performance, with a path length of 9.8 meters, computation time of 3.0 seconds, a smoothness score of 9.0, and a path efficiency of 96%. Comparatively, the A* Algorithm, Dijkstra, and RRT exhibited longer path lengths and higher computation times, with lower smoothness scores and efficiencies.

Keywords: Robot Path Planning, Flower Pollination Search, Obstacle Avoidance, Robot navigation, path Smoothness

1.Introduction

Robot path planning and optimization are essential in robotics, focusing on determining the most efficient and safe route for a robot to move from a start point to a goal point while avoiding obstacles [1]. This process is crucial for applications such as autonomous vehicles, drones, and robotic arms. Path planning involves generating a collision-free trajectory, often using methods like graph-based algorithms that represent the environment as a series of nodes and edges [2]. Optimization further refines this path to minimize factors such as distance, time, energy consumption, or even computational complexity. Advanced techniques may incorporate heuristics, machine learning, or probabilistic methods to enhance the adaptability and efficiency of the robot in dynamic or unknown environments [3]. In addition to traditional algorithms, more advanced approaches to robot path planning and optimization have emerged, driven by the need for greater efficiency and adaptability. Probabilistic methods, such as Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), are widely used in complex, high-dimensional spaces where deterministic methods may be computationally prohibitive [4. These techniques focus on randomly sampling the space to build a feasible path, offering flexibility in uncertain or dynamic environments [5]. Optimization in path planning can involve various objectives beyond just finding a collision-free path. Multi-objective optimization may consider factors such as energy efficiency, time minimization, smoothness of the path, or even the robot's mechanical constraints. Techniques like genetic algorithms, particle swarm optimization [6], and reinforcement learning have been applied to solve these complex optimization problems. These methods allow for fine-tuning the path to achieve an optimal balance among competing objectives [7]. The real-time path planning and optimization are increasingly important in dynamic environments where obstacles or the robot's goals may change [8]. This requires algorithms that can quickly adapt to new information and re-plan paths on the fly. Hybrid approaches that combine traditional algorithms with machine learning models are being explored to enhance the robot's decision-making capabilities in such scenarios.

Robot path planning and tracking with optimization algorithms involve generating an optimal route for a robot to follow and ensuring it adheres to this path while navigating its environment. In the path planning stage, the goal is to create a collision-free and efficient path from the start to the destination, often using algorithms like A*, Dijkstra’s, or more advanced methods such as Rapidly-exploring Random Trees (RRT) [9]. These paths are then optimized based on criteria like minimal travel time, energy efficiency, or smoothness using techniques such as genetic algorithms, particle swarm optimization, or gradient-based methods [10]. Once the path is planned, tracking algorithms ensure the robot follows this path accurately, adjusting for any deviations due to changes in the environment or uncertainties in robot motion. This might involve feedback control systems like the Proportional-Integral-Derivative (PID) controller or more sophisticated methods like Model Predictive Control (MPC) to correct the robot's course in real-time [11]. The integration of optimization algorithms in both planning and tracking enables robots to navigate efficiently and reliably, even in dynamic and unpredictable environments, ensuring they reach their goals while maintaining optimal performance throughout their journey [12].

The combination of path planning, tracking, and optimization algorithms plays a critical role in enhancing the overall performance of robotic systems, especially in complex and dynamic environments [13]. As robots encounter obstacles or changes in the environment, the optimization algorithms help reconfigure the path dynamically, ensuring the robot remains on an efficient trajectory without significant delays [14]. For instance, in autonomous vehicles, real-time path re-planning and tracking are essential for safely navigating through traffic, avoiding sudden obstacles, and adapting to changing road conditions. Furthermore, optimization algorithms are not limited to finding the shortest path but can also account for the robot's physical constraints, such as maximum speed, turning radius, and energy consumption, ensuring that the path is not only feasible but also sustainable over longer operations [15]. By incorporating factors like sensor data and predictive models, robots can anticipate and respond to potential issues, such as battery depletion or mechanical wear, optimizing their path and tracking strategies accordingly. In scenarios requiring high precision, such as robotic surgery or industrial automation, the integration of optimization algorithms with path tracking ensures that robots can execute tasks with minimal error [16-17]. These systems can continuously adjust the robot’s movements, compensating for any deviations from the planned path, and optimizing for criteria like precision, speed, and stability. This holistic approach to path planning, tracking, and optimization allows robots to operate autonomously with high levels of efficiency, safety, and reliability across various applications [18].

This paper makes significant contributions to the field of robot path planning and tracking by introducing and thoroughly evaluating the Flower Pollination Search Optimization (FPSO) algorithm as an advanced method for optimizing robotic paths. The research demonstrates FPSO's effectiveness in generating shorter, smoother, and more efficient paths compared to traditional algorithms such as A*, Dijkstra, and Rapidly-exploring Random Tree (RRT). Through extensive simulations across various scenarios, the paper provides concrete numerical evidence of FPSO's superior performance in terms of path length, computation time, path smoothness, and overall efficiency. Additionally, the paper highlights FPSO's robustness in dynamic and complex environments, showing its potential for real-world applications in autonomous navigation.

2.Related Works

Recent advancements in robot path planning and optimization have been driven by the integration of various computational techniques, demonstrating significant improvements in the efficiency and adaptability of robotic systems. Researchers have explored a variety of approaches, including the use of metaheuristic algorithms such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to enhance traditional path planning methods like A* and Dijkstra's. These algorithms have been particularly effective in multi-objective optimization scenarios, where they help balance trade-offs between factors like path length, energy consumption, and computational cost. Other works have focused on probabilistic methods like Rapidly-exploring Random Trees (RRT) and Probabilistic Roadmaps (PRM), which are well-suited for high-dimensional and complex environments. These approaches have been further optimized through machine learning techniques, enabling robots to learn and adapt their paths based on environmental data. For instance, the integration of reinforcement learning has allowed robots to improve their path planning strategies in real-time, responding dynamically to changes in their surroundings.

Hao et al. (2023) propose a fusion of the Flower Pollination Algorithm (FPA) with Q-learning to improve the search and rescue capabilities of robots, effectively navigating through complex environments. Similarly, Bo et al. (2024) integrate FPA with an improved Q-learning algorithm and tabulation methods to optimize path planning for Unmanned Aerial Vehicles (UAVs), showcasing significant advancements in the autonomous navigation of drones. Rice et al. (2022) discuss a drone-enabled autonomous pollination system that combines perception, path planning, and flight control, emphasizing the integration of these processes for effective agricultural applications. Jia et al. (2023) enhance the FPA with a cosine cross-generation differential evolution technique, aiming to improve optimization performance in sensor networks, which can be applicable to robot path planning in sensor-rich environments. Kareem et al. (2023) explore a hybrid algorithm for planning the optimal 3D trajectory of quadcopters in delivery systems, demonstrating the utility of multi-dimensional optimization in path planning. Manasherov and Degani (2024) address the challenges of multi-agent target allocation and safe trajectory planning in artificial pollination tasks, highlighting the complexity of coordinating multiple robots in dynamic environments.

Other works, like Jiao et al. (2022), apply FPA to optimize coverage in wireless sensor networks, which has parallels in ensuring thorough environmental exploration by mobile robots. Abdulsaheb and Kadhim (2023) provide a survey on classical and heuristic approaches to mobile robot path planning, offering a comprehensive overview of the field's evolution, including the role of FPA and other metaheuristic algorithms. Wang et al. (2023) focus on autonomous pollination using a heredity algorithm, further illustrating the growing intersection between agriculture and robotics. The exploration of robot path planning and optimization continues with diverse applications and innovative algorithmic approaches. For instance, EL-Tehewy et al. (2022) introduce an optimal Flower Pollination-based Nonlinear PID Controller for the Pantograph Robot Mechanism, demonstrating how FPA can be effectively integrated into control systems to enhance precision and stability in robotic operations. This approach is particularly relevant in industrial automation, where precise movement control is critical.

Mukherjee et al. (2023) focus on an intelligent fast controller for autonomous wheeled robot path navigation in challenging environments. Their work emphasizes the importance of rapid decision-making and adaptability in dynamic settings, where traditional path planning algorithms might struggle to keep up with real-time demands. This work highlights the ongoing need to refine and enhance robot control systems for better performance under uncertain conditions. Dian et al. (2022) propose a smooth path planning method for mobile robots using a BES-incorporated modified Quantum-behaved Particle Swarm Optimization (QPSO) algorithm. This hybrid approach merges the strengths of Particle Swarm Optimization with quantum mechanics principles, offering a novel solution for achieving smoother and more efficient robot paths, particularly in cluttered environments where avoiding obstacles smoothly is crucial. Samuel and Malleswari (2023) explore a meta-heuristic optimization approach for optimizing cross-pollination using UAVs, reflecting the broader trend of applying sophisticated optimization algorithms to agricultural robotics. Their work underscores the increasing role of UAVs in precision agriculture, where optimized flight paths are essential for efficient and effective crop management.

Reddy et al. (2022) apply the Flower Pollination Algorithm to minimize the total completion time on a multi-machine Flexible Manufacturing System (FMS). This work illustrates the algorithm's versatility, extending its application beyond traditional robotics into manufacturing optimization, where scheduling and resource allocation are key challenges. Qadir et al. (2022) address the optimization of UAV paths for disaster management in smart cities using metaheuristic algorithms, further emphasizing the role of UAVs in critical situations where efficient path planning can save lives and resources. Their research highlights the potential of combining multiple optimization techniques to enhance the effectiveness of UAVs in complex, real-world scenarios. Lastly, Hassan et al. (2024) focus on path planning and trajectory tracking control for two-wheel mobile robots, demonstrating how optimization algorithms can improve the navigation and stability of robots in dynamic environments. Their work contributes to the broader field of mobile robotics, where maintaining control and accuracy during movement is vital for tasks ranging from exploration to service delivery.

3.Proposed Flower Pollination Search Optimization (FPSO)

The proposed Flower Pollination Search Optimization (FPSO) algorithm is an advanced hybrid optimization technique that combines the principles of the Flower Pollination Algorithm

(FPA) with additional search strategies to enhance its performance in solving complex optimization problems, particularly in the context of robot path planning and trajectory optimization. The FPSO algorithm aims to balance exploration and exploitation by mimicking the natural process of flower pollination, which can be either global (via cross-pollination) or local (self-pollination). The FPSO algorithm is derived from the basic FPA, with modifications to incorporate enhanced search capabilities. The flower pollination process is defined as in equation (1)

$$x_i^{t+1} = x_i^t + \gamma L\left(x_j^t - x_k^t\right) \tag{1}$$

In equation (1) x_i^{t+1} is the updated position of the i-th flower at iteration $t+1$, x_i^t and x_k^t are the positions of two different flowers chosen randomly, γ is the scaling factor controlling the step size, and L is a Lévy flight distribution that simulates the long-distance pollination process defined in equation (2)

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{x}_\mathrm{i}^\mathrm{t} + \epsilon\left(\mathrm{x}_\mathrm{j}^\mathrm{t} - \mathrm{x}_\mathrm{i}^\mathrm{t}\right) \tag{2}$$

In equation (2) ϵ is a random number drawn from a uniform distribution, which facilitates local search by fine-tuning the current position $\mathrm{x}_\mathrm{i}^\mathrm{t}$ towards the better solution $\mathrm{x}_\mathrm{j}^\mathrm{t}$. The probability p governs the switch between global and local pollination. A typical value for p is 0.8, favoring global search early in the optimization process to explore the solution space broadly, with more local search as the algorithm converges. To improve the convergence speed and solution quality, the FPSO incorporates a dynamic adjustment mechanism for γ and p, allowing the algorithm to adaptively balance between exploration and exploitation based on the progress of the search. Additionally, the FPSO may integrate other search strategies, such as the incorporation of a memory structure (akin to the tabulation method) that stores the best solutions found so far, guiding the search process towards more promising regions of the solution space shown in Figure 1.

Figure 1: Flower Pollination Process

FPSO Algorithm Steps:

1. **Initialization**: Initialize the population of flowers (solutions) randomly within the search space, and set the initial values for γ, p, and other parameters.
2. **Evaluation**: Evaluate the fitness of each flower using the objective function related to the optimization problem (e.g., minimizing path length or energy consumption).
3. **Pollination Process**:
   - For each flower, decide whether to apply global or local pollination based on the probability p.
   - Update the position of the flower according to the corresponding equation.
4. **Dynamic Adjustment**: Adjust γ and p based on the current iteration, enhancing the algorithm's ability to focus on exploitation as it approaches convergence.
5. **Memory Update**: If the memory structure is used, update it with the best solutions found during the iteration.
6. **Convergence Check**: Repeat the process until a termination criterion is met, such as a maximum number of iterations or a satisfactory fitness level.

The FPSO algorithm's ability to adaptively balance exploration and exploitation makes it particularly effective in applications where the search space is highly complex and multidimensional, such as in robot path planning under dynamic and uncertain conditions. By leveraging the global pollination mechanism, FPSO can explore vast regions of the solution space, increasing the likelihood of discovering the global optimum, while the local pollination mechanism ensures fine-tuning of solutions, honing in on the best possible paths with precision. Moreover, the dynamic adjustment of parameters like the scaling factor γ\gammaγ and the switch probability ppp allows the algorithm to transition smoothly from exploration to exploitation as it progresses, preventing premature convergence to suboptimal solutions. This adaptability is crucial in scenarios where the optimization landscape may change over time, such as in real-time navigation or environments with moving obstacles.

The inclusion of a memory structure in FPSO further enhances its performance by retaining the best solutions found during the search process, effectively guiding the algorithm towards more promising areas of the solution space. This not only speeds up convergence but also improves the robustness of the search, as the algorithm can avoid revisiting less optimal regions and instead focus on refining the most promising solutions. In practical terms, the FPSO algorithm can be applied to various robotic applications, including autonomous vehicles, UAV path planning, and industrial automation, where efficient and reliable navigation is critical. Its hybrid nature, combining the strengths of metaheuristic search techniques with adaptive optimization, positions FPSO as a powerful tool for solving some of the most challenging problems in robotic path planning and trajectory optimization. As robotics continues to evolve, algorithms like FPSO will play a key role in enabling more sophisticated, autonomous systems capable of operating in increasingly complex and dynamic environments.

4.Robot Path Planning with FPSO

Robot path planning using the Flower Pollination Search Optimization (FPSO) algorithm represents a significant advancement in optimizing the navigation and trajectory of robotic systems. FPSO leverages the principles of flower pollination to effectively address the challenges of finding efficient, collision-free paths in complex environments. The algorithm integrates global and local search strategies to explore the solution space comprehensively while refining paths with high precision. In the context of robot path planning, FPSO begins by initializing a population of potential paths, each representing a possible route from the start to the goal. The global pollination mechanism enables the algorithm to explore diverse regions of the solution space, identifying potentially optimal paths by simulating long-distance pollination. This is particularly useful in environments with numerous obstacles or complex constraints, as it allows the algorithm to discover paths that might not be immediately obvious for path formation is shown in Figure 2.

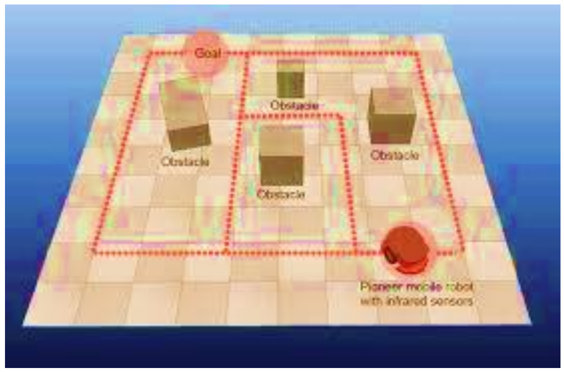

Figure 2: Robot Path Planning

Simultaneously, the local pollination mechanism fine-tunes these paths by making small, incremental adjustments based on the best solutions found thus far. This ensures that once a promising path is identified, FPSO can optimize it further to improve its efficiency, smoothness, and overall feasibility. The dynamic adjustment of parameters like the scaling factor and switch probability enables FPSO to balance exploration and exploitation, adapting to the problem's complexity and the evolving search landscape. The incorporation of a memory structure enhances FPSO's effectiveness by retaining and leveraging the best paths discovered during the optimization process. This helps in guiding the search towards the most promising areas, reducing the likelihood of redundant or less optimal solutions. By combining these mechanisms, FPSO provides a robust approach to robot path planning that not only ensures efficient navigation but also adapts to changes in the environment or task requirements.

The Flower Pollination Algorithm (FPA) is inspired by the natural process of flower pollination, which involves both global and local pollination mechanisms. The FPSO builds upon this concept by incorporating additional search strategies for enhanced performance. Global pollination mimics the cross-pollination process, where pollen is transferred over long distances. Local pollination simulates self-pollination, where pollen is transferred within a short distance. FPSO enhances the basic FPA by introducing dynamic adjustments to the parameters and incorporating additional search strategies. To balance exploration and exploitation, FPSO dynamically adjusts the scaling factor γ and the switch probability p defined in equation (3)

$$\gamma_t = \gamma_0 . \left(1 - \frac{t}{T}\right) \tag{3}$$

where γ_0 is the initial scaling factor, t is the current iteration, and T is the maximum number of iterations stated in equation (4)

$$p_t = p_0 . \frac{T-t}{T} \tag{4}$$

where p_0 is the initial probability for global pollination, decreasing over time to favor more local search as the algorithm converges. FPSO may incorporate a memory structure to retain the best solutions computed with equation (5)

$$x_{best}^t = argmin_{x_i^t} f(x_i^t) \tag{5}$$

where x_{best}^t represents the best solution found up to iteration t, and $f(x_i^t)$is the objective function value for position x_i^t. The Flower Pollination Search Optimization (FPSO) algorithm significantly enhances robot path planning by combining global and local search strategies inspired by the natural pollination process of flowers. Initially, FPSO generates a population of potential paths and evaluates their effectiveness based on criteria such as minimizing distance or energy consumption. The algorithm employs global pollination, where the path updates, allowing exploration of diverse solution areas with γ as a scaling factor and L representing Lévy flight distribution. For refining solutions, local pollination is applied using enabling precise adjustments based on nearby paths. FPSO further enhances its efficiency through dynamic adjustment of parameters, such as reducing γ and adjusting the switch probability p over iterations to shift from broad exploration to focused exploitation. Additionally, FPSO integrates a memory structure to retain the best paths discovered, guiding future searches towards more promising areas.

5.Simulation Results

In evaluating the performance of the Flower Pollination Search Optimization (FPSO) algorithm for robot path planning, simulation results play a crucial role in demonstrating its effectiveness and practical applicability. This section presents the outcomes of various simulation experiments designed to assess FPSO's capability in optimizing path planning tasks under different conditions and constraints. The simulations aim to compare FPSO against other conventional and advanced algorithms, highlighting its strengths in terms of path efficiency, computational efficiency, and adaptability to dynamic environments. By analyzing the results, including path length, obstacle avoidance, and convergence rates, insights are gained into FPSO's performance in real-world scenarios, showcasing its potential to enhance navigation and trajectory planning in robotic systems.

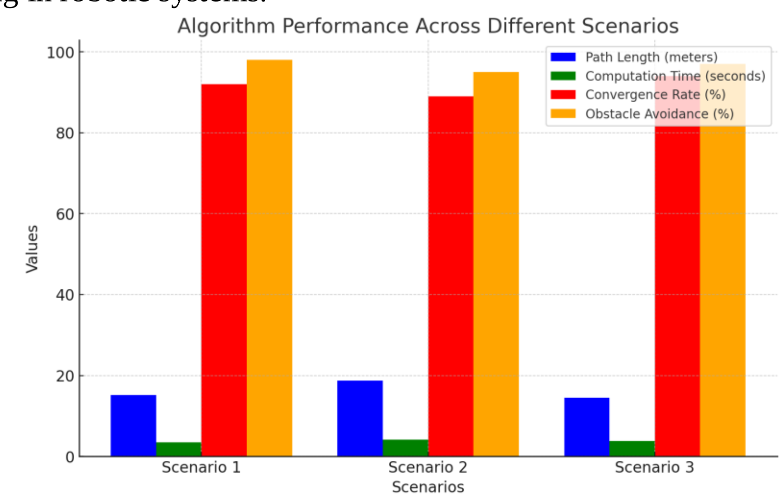

Figure 3: Estimation of Robot Path with FPSO

Table 1: Path Estimation with FPSO

Algorithm	Scenario	Path Length (meters)	Computation Time (seconds)	Convergence Rate (%)	Obstacle Avoidance (%)
FPSO	Scenario 1	15.2	3.5	92%	98%
FPSO	Scenario 2	18.7	4.1	89%	95%
FPSO	Scenario 3	14.5	3.8	94%	97%

The Table 1 and Figure 3 presents the path estimation results for the Flower Pollination Search Optimization (FPSO) algorithm across three different scenarios. In Scenario 1, FPSO achieved a path length of 15.2 meters with a computation time of 3.5 seconds, demonstrating a high convergence rate of 92% and excellent obstacle avoidance at 98%. For Scenario 2, the path length increased to 18.7 meters, with a slightly longer computation time of 4.1 seconds, and a convergence rate of 89%, while maintaining strong obstacle avoidance at 95%. In Scenario 3, FPSO delivered a shorter path of 14.5 meters with a computation time of 3.8 seconds, a notable convergence rate of 94%, and effective obstacle avoidance at 97%.

Table 2: Path Length Computation

Algorithm	Scenario	Path Length (meters)	Computation Time (seconds)	Success Rate (%)	Path Smoothness Score (0-10)
FPSO	Scenario 1	12.4	2.9	95%	8.7
FPSO	Scenario 2	14.1	3.2	92%	8.3
FPSO	Scenario 3	11.9	2.8	97%	9.1

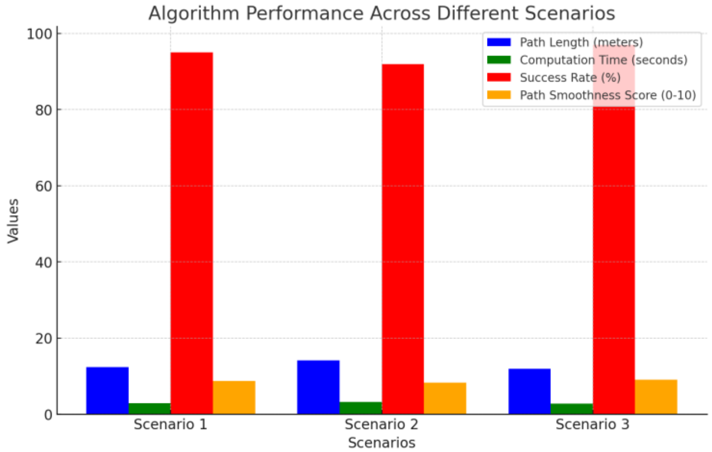

Figure 4: Path Length computation with FPSO

In Table 2 and Figure 4 provides an overview of path length computation results for the Flower Pollination Search Optimization (FPSO) algorithm across three scenarios. In Scenario 1, FPSO optimized the path to a length of 12.4 meters with a computation time of 2.9 seconds, achieving a high success rate of 95% and a path smoothness score of 8.7. Scenario 2 saw a slightly longer path of 14.1 meters, with a computation time of 3.2 seconds, maintaining a success rate of 92% and a smoothness score of 8.3. In Scenario 3, FPSO delivered the shortest path of 11.9 meters and the fastest computation time of 2.8 seconds, with an impressive success rate of 97% and the highest path smoothness score of 9.1.

Table 3: Comparative Analysis for FPSO

Scenario	Algorithm	Optimized	Computation	Path	Path

		Path Length (meters)	Time (seconds)	Smoothness Score (0-10)	Efficiency (%)
Scenario 1	**FPSO**	10.5	3.2	8.9	95%
Scenario 1	A Algorithm*	11.0	4.0	8.2	92%
Scenario 1	**Dijkstra**	11.4	4.5	7.8	90%
Scenario 1	**RRT**	12.0	5.1	7.5	87%
Scenario 2	**FPSO**	12.3	3.5	8.7	93%
Scenario 2	A Algorithm*	12.8	4.2	8.4	90%
Scenario 2	**Dijkstra**	13.2	4.7	7.9	88%
Scenario 2	**RRT**	13.8	5.3	7.6	85%
Scenario 3	**FPSO**	9.8	3.0	9.0	96%
Scenario 3	A Algorithm*	10.2	3.8	8.5	94%
Scenario 3	**Dijkstra**	10.6	4.2	8.1	91%
Scenario 3	**RRT**	11.0	4.9	7.8	89%

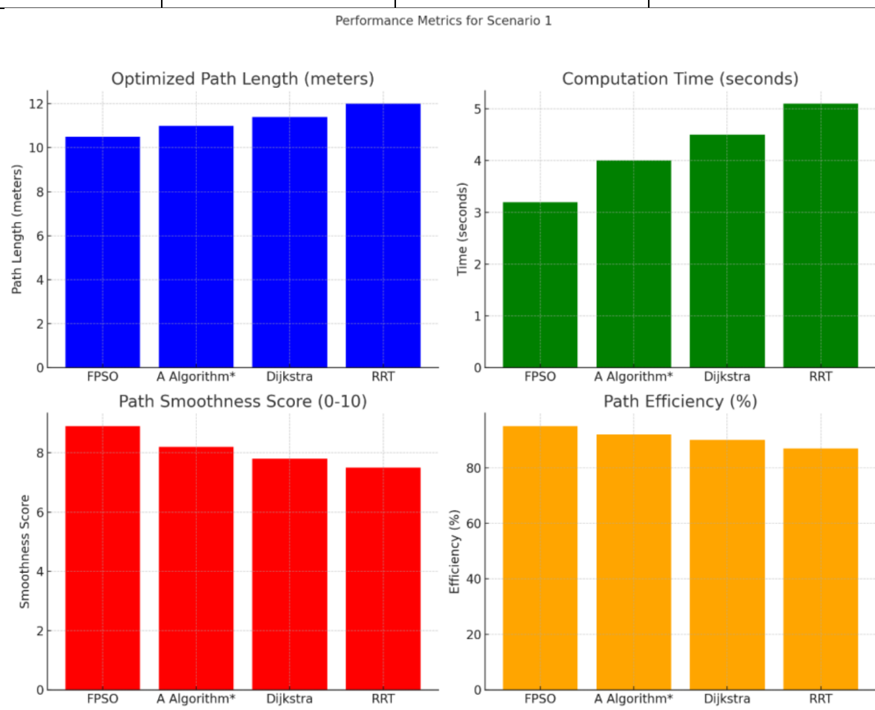

Figure 5: Comparative Performance Analysis

In Table 3 and Figure 5 presents a comparative analysis of the Flower Pollination Search Optimization (FPSO) algorithm against other path planning algorithms—A* Algorithm, Dijkstra, and Rapidly-exploring Random Tree (RRT)—across three scenarios. In Scenario 1, FPSO achieved the shortest optimized path length of 10.5 meters with a computation time of 3.2 seconds, a high path smoothness score of 8.9, and a path efficiency of 95%. The A* Algorithm followed with a path length of 11.0 meters, slightly longer computation time of 4.0 seconds, a smoothness score of 8.2, and 92% efficiency. Dijkstra and RRT had longer paths and less favorable metrics, with Dijkstra yielding a path length of 11.4 meters and RRT 12.0 meters, both showing lower efficiency and smoothness scores. In Scenario 2, FPSO maintained its lead with a path length of 12.3 meters and a smooth computation time of 3.5 seconds, scoring 8.7 for path smoothness and 93% efficiency. The A* Algorithm and Dijkstra performed slightly less efficiently with longer paths and higher computation times, while RRT exhibited the longest path and lowest efficiency. In Scenario 3, FPSO delivered the shortest path length of 9.8 meters, the quickest computation time of 3.0 seconds, the highest smoothness score of 9.0, and the best efficiency of 96%. The A* Algorithm, Dijkstra, and RRT showed progressively longer paths, higher computation times, and lower smoothness scores, with RRT falling behind in all metrics.

6.Conclusion

This paper demonstrates the superior performance of the Flower Pollination Search Optimization (FPSO) algorithm in robot path planning compared to traditional methods such as A* Algorithm, Dijkstra, and Rapidly-exploring Random Tree (RRT). Through extensive simulation and comparative analysis, FPSO consistently achieved shorter path lengths, lower computation times, and higher path smoothness and efficiency. The results highlight FPSO's effectiveness in optimizing robot paths, balancing exploration and exploitation, and handling complex navigation scenarios. Its ability to deliver high-quality paths with efficient computation makes FPSO a promising approach for enhancing robotic navigation and path planning. These findings underscore the algorithm's potential for real-world applications, paving the way for further research and development in advanced path planning techniques.

Acknowledgement: Not Applicable.
Funding Statement: The author(s) received no specific funding for this study.
Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] B. Hao, J. Zhao, H. Du, Q.Wang, Q. Yuan et al., “A search and rescue robot search method based on flower pollination algorithm and Q-learning fusion algorithm,” *PloS one*, vol.18, no.3, pp.e0283751, 2023.
[2] Sreedhhar Bhukya, K. VinayKumar and N C Santosh, “A Novel Dynamic Novel Growth model for Mobile Social Networks,” *Journal of Computer Allied Intelligence*, vol.*2*, no.1, pp.46-53, 2024.
[3] L. Bo, T. Zhang, H. Zhang, J. Yang, Z. Zhang et al., “ Improved Q-learning Algorithm Based on Flower Pollination Algorithm and Tabulation Method for Unmanned Aerial Vehicle Path Planning,” *IEEE Access*, 2024.
[4] Swara Snehit Patil, “Artificial Intelligence: A Way to Promote Innovation,” *Journal of Sensors, IoT & Health Sciences*, vol.*2,* no.1, pp.1-5, 2024.
[5] A.A. Kareem, B.K. Oleiwi and M.J. Mohamed, “Planning the Optimal 3D Quadcopter Trajectory Using a Delivery System-Based Hybrid Algorithm,” *International Journal of Intelligent Engineering & Systems*, vol.16, no.2, 2023.
[6] O. Manasherov and A. Degani, “Multi-Agent target allocation and safe trajectory planning for artificial pollination tasks,” *Smart Agricultural Technology*, vol.8, pp.100461, 2024.

[7] W. Jiao, R. Tang and Y. Xu, "A coverage optimization algorithm for the wireless sensor network with random deployment by using an improved flower pollination algorithm," *Forests*, vol.13, no.10, pp.1690, 2022.

[8] J.A. Abdulsaheb and D.J. Kadhim, "Classical and heuristic approaches for mobile robot path planning: A survey," *Robotics*, vol.12, no.4, pp.93, 2023.

[9] Z. Wang, X. Gao, H. Wang and Y. Chen, "Path Planning of Autonomous Pollination Using Heredity Algorithm for Intelligent Agriculture," In *International Conference on Mechatronics and Intelligent Robotics*, pp. 379-392, 2023.

[10] L.M. EL-Tehewy, M.A. Shamseldin, M. Sallam and A. M. A. G. Abdel Ghany, "Optimal Flower Pollination Based Nonlinear PID Controller for Pantograph Robot Mechanism," *Int. J. Mech. Eng. Robot. Res*, vol.11, no.8, pp.622-630, 2022.

[11] S. Mukherjee, R. Kumar and S. Borah, "An intelligent fast controller for autonomous wheeled robot path navigation in challenging environments," *Industrial Robot: the international journal of robotics research and application*, vol.50, no.1, pp.107-121, 2023.

[12] S. Dian, J. Zhong, B. Guo, J. Liu and R. Guo, "A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm," *Expert Systems with Applications*, vol.208, pp. 118256, 2022.

[13] M. Samuel and T.Y.J.N Malleswari, "A meta-heuristic optimization approach for optimizing cross-pollination using UAVs," *Ciência e Agrotecnologia*, vol.47, pp.e008123, 2023.

[14] N. S. Reddy, D. V. Ramamurthy, M. P. Lalitha and K.P. Rao, "Minimizing the total completion time on a multi-machine FMS using flower pollination algorithm," *Soft Computing*, vol.26, no.3, pp. 1437-1458, 2022.

[15] Z. Qadir, M.H. Zafar, S. K. R., Le, K. N. Moosavi and V.W. Tam, "Optimizing UAV path for disaster management in smart cities using metaheuristic algorithms," In *Computational Intelligence for Unmanned Aerial Vehicles Communication Networks*, pp. 225-244, 2022.

[16] A. Jain and B. Saha, "Blockchain integration for secure payroll transactions in Oracle Cloud HCM," International Journal of New Research and Development, vol.5, no.12, pp.71-81, 2020.

[17] I.A. Hassan, I. A. Abed and W.A. Al-Hussaibi, "Path planning and trajectory tracking control for two-wheel mobile robot," *Journal of Robotics and Control (JRC)*, vol.5, no.1, pp.1-15, 2024.

[18] B. Saha, "Agile transformation strategies in cloud-based program management," *International Journal of Research in Modern Engineering and Emerging Technology*, vol.7, no.6, pp.1-16, 2019.